Near real-time processing of proteomics data using Hadoop.

Hillman, Chris; Ahmad, Yasmeen; Whitehorn, Mark; Cobley, Andy

Published in:
Big Data

DOI:
10.1089/big.2013.0036

Publication date:
2014

ORIGINAL ARTICLE

NEAR REAL-TIME PROCESSING OF PROTEOMICS DATA USING HADOOP

Chris Hillman,[1] Yasmeen Ahmad,[2] Mark Whitehorn,[1] and Andy Cobley[1]

Abstract

This article presents a near real-time processing solution using MapReduce and Hadoop. The solution is aimed at some of the data management and processing challenges facing the life sciences community. Research into genes and their product proteins generates huge volumes of data that must be extensively preprocessed before any biological insight can be gained. In order to carry out this processing in a timely manner, we have investigated the use of techniques from the big data field. These are applied specifically to process data resulting from mass spectrometers in the course of proteomic experiments. Here we present methods of handling the raw data in Hadoop, and then we investigate a process for preprocessing the data using Java code and the MapReduce framework to identify 2D and 3D peaks.

Introduction

The human genome project was one of the largest and most well-known scientific endeavors of recent times. This project characterized the entire set of genes found in human DNA. Following on from this, the focus has now moved to studying proteins, which are the products of genes found in cells. Genes may act as a blueprint, but it is in fact the proteins in a cell that carry out functions. Proteins are the building blocks of cells and their study is very important in the research of disease. Proteomics can be defined as the large-scale study of protein properties, such as expression levels, modifications, and interactions with other proteins. By studying proteins and their properties, it is possible to gain a deeper understanding of how proteins should function in healthy cells compared with diseased cells.[1] In a typical proteomics experiment, an instrument called a mass spectrometer may be used to identify proteins and measure their quantities. However, because of the sensitivity of the instruments, a protein molecule is in fact too large to be identified. Hence, proteins must be broken down into smaller fragments called peptides. The mix of peptides is fed into a column before entering the mass spectrometer. Some of the peptides pass through the column relatively rapidly, and others effectively bind slightly to the material in the column with the result that their passage through the column is slowed as they are retained for a while in the column. Mass spectrometers measure the mass, charge, and retention time of particles—in this case, the peptides. This data is captured as the mass-to-charge ratio and quantity (measured as the "intensity") of the peptide molecules. It is typically presented as a graphical output (Fig. 1). Within each spectrum the data points form Gaussian curves, with each curve representing a peptide identification. Hence, the first step of computational analysis is to identify these curves, known as 2D peaks. The next step of the process takes into account the time element of the instrument analysis. When a sample is submitted to an instrument, it can take between 100 minutes to 4 hours for the sample to be fully analyzed. The sample passes through the instrument, with smaller peptide molecules passing through first, ranging all the way up to the largest peptide molecules at the end. A spectrum (Fig. 1) is taken every second as a

[1]School of Computing and [2]Centre for Gene Regulation & Expression, University of Dundee, Nethergate, Dundee, United Kingdom.

BIG DATA MARCH 2014 • DOI: 10.1089/big.2013.0036

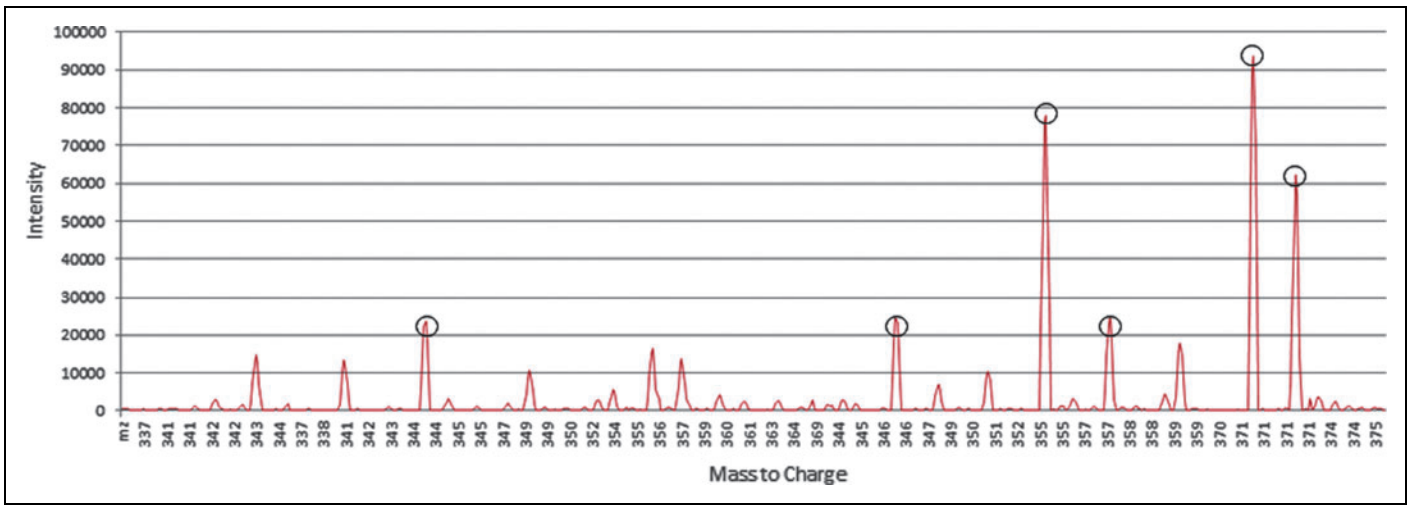

FIG. 1. Scan from a mass spectrometer with several 2D peaks identified.

snapshot documenting what is present in the instrument at that point in time. Each peptide may be seen in several snapshots before it finishes passing through the instrument. Hence, spectra need to be joined together in a third dimension—time. This means that the 2D peaks now form 3D peaks, which must be computationally detected. An added complication in the spectra arises from the fact that the same peptide may be observed in nature with different masses. This is caused by the natural occurrence of different forms of carbon, which make up a peptide molecule. However, these multiple forms of a peptide are predictable and form a pattern in the spectra (Fig. 2). This pattern is known as an isotopic envelope. As all peaks within this envelope belong to the same peptide, they must be aggregated.

Any given experiment in proteomics will often consist of multiple samples. The mass spectrometry instrumentation produces a RAW data file for each sample processed. As the number of samples and the individual sample complexity increase, the amount of data also increases proportionally. A typical RAW file output from a Thermo Orbitrap mass spectrometer, as used by the University of Dundee, contains approximately 40,000 scans and each scan contains approximately 20,000 data points. This results in any single dataset from an experiment comprising up to 800,000,000 data points. Furthermore, in the course of an experiment it is usual to create several technical and biological replicates, which all add to the volume of data generated. A large laboratory with 10 instruments in use could routinely produce billions of data points per day. Current processing methods using desktop computers would struggle to keep up with this volume of data as it is produced. This leads to a backlog of data processing as the time taken to process data exceeds the time taken for the next experiment to run. The aim of this research is to produce a system based on a horizontally scalable architecture that, with the addition of computing nodes, will allow the processing time to remain constant as the processing requirements increase.

"CURRENT PROCESSING METHODS USING DESKTOP COMPUTERS WOULD STRUGGLE TO KEEP UP WITH THIS VOLUME OF DATA AS IT IS PRODUCED."

Hadoop consists of Hadoop distributed file system (HDFS), a distributed fault-tolerant file system, and MapReduce, a framework used to distribute processing across clusters of computers, first proposed by engineers working for Google.[2] It is designed to be run on clusters of commodity hardware and frees the programmer from the complexity of allocating, monitoring, and running many parallel tasks. Apache Hadoop is an open-source implementation that is being widely evaluated and adopted by the business community as an alternative to traditional relational databases in terms of storage and the types of analysis that can be performed.[3] In reviewing the current literature, there are very few references to the use of MapReduce-style parallelization for the processing of data created in the course of proteomic experiments. There is reference to this paradigm being used to show the possibility of identifying proteins by matching peak lists to theoretical databases[4–6] although these articles do not detail the MapReduce steps process involved in the peak-list production. Our research is concerned with the investigation of using parallel compute clusters and programming methods to develop efficient techniques for dealing with larger data volumes and complexity while producing results in a reasonable time frame. The specific algorithms used to

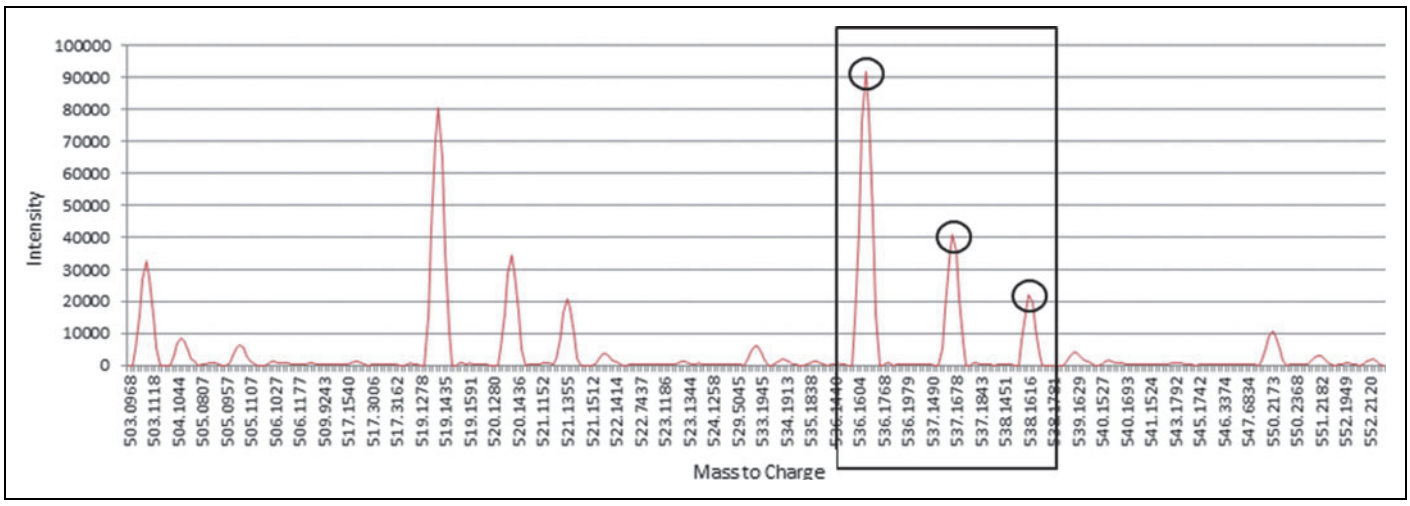

FIG. 2. Peaks within an isotopic envelope.

pick the 2D and 3D peaks are a Java representation of algorithms developed at the University of Dundee. The research detailing these algorithms is unpublished at the time of writing this article and cannot be referenced. All of the output from the MapReduce implementation has been extensively validated against the output used in the development of the new peak detection techniques and found to be accurate. Using the MapReduce code on a horizontally scalable architecture, we expect to approach real-time output of processed data. In this context, real time will be measured as being within minutes as opposed to the many hours or days that current processing methods take.

Data Processing

The output from a mass spectrometer will typically be a RAW file in a vendor-specific binary format; the convention is to convert these files into an XML format called mzML, which was defined by the proteomics community.[7] The files used in the course of this work were converted using the msconvert tool, part of the proteowizard tool set. The mzML standard is designed for the storage of data and transfer between researchers and institutions in a vendor- and platform-independent format. It is acknowledged that conversion to this format is necessary whatever the final processing steps taken. Conversion to mzML from Thermo RAW files requires a vendor-provided dll file that can be used only on a Windows platform. The Windows PC–based conversion process using msconvert takes several minutes to perform, which is a serial process constant and not considered further in the parallel-processing evaluation.

As files are loaded into the HDFS file system, the default Hadoop behavior to split files into 64 Mb chunks to the nearest newline character is likely to result in an XML file being split between opening and closing tags. To avoid any complexity of processing large (7 Gb+) XML files, a new file format was investigated that allows the data to be distributed and scans to be processed in parallel rather than sequentially.

XML file conversion

The XML data consists of a header section containing information about the environment in which the experiment was performed, for example, machine settings, researcher details, date, and time. Subsequently, there is a section containing information about each scan performed with the data from the scan held as two Base64-encoded strings, one for the mass-to-charge ratio and one for the intensity of the particles detected. In order to create a new file format holding just the information relevant for detecting the peptides (and hence the proteins) present in the sample, two methods of XML parsing were investigated. The first is called document object model (DOM), which is unsuitable for large files as all data is read into memory before processing. The second method investigated was the SAX parser or Simple API for XML. This parser processes the input file in a sequential manner and does not attempt to read the entire file into memory. The SAX parser library is very simple to use: a convertor was written and compiled, and conversion between mzML and the new tab-delimited file format tested and timed. This has proved remarkably successful: an initial 5.4 Gb mzML file can be converted into a tab-delimited flat file format in under 3 minutes using a midlevel laptop (quadcore i5 processor@ 2.5 Ghz and a solid-state drive). This puts the conversion time from a 5.4 Gb vendor-specific RAW file to mzML and finally to tab-delimited format at approximately 10 minutes. The new file format contains the following data as tab-delimited columns: scan number, mslevel, retention time, m/z Base64 array, intensity Base64 array, precursor ion m/z, precursor ion intensity, and precursor ion charge. To be processed in Hadoop, the data needs to be moved into the Hadoop file system; the default method is to copy the data

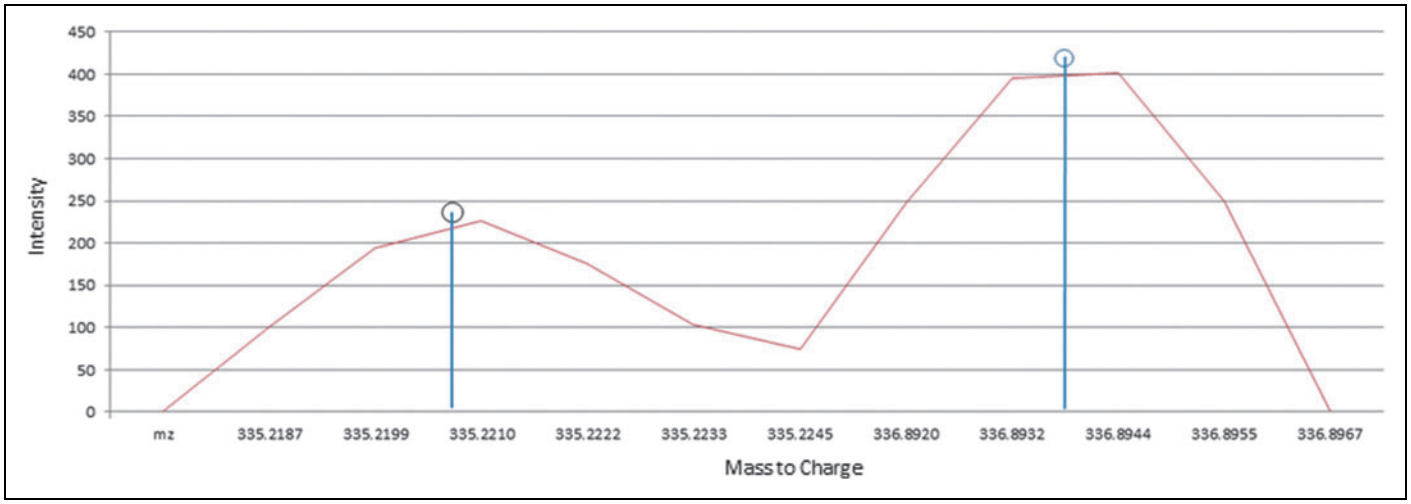

FIG. 3. Overlapping curves with theoretical peaks identified.

onto a Linux server and then use the Hadoop–fs put command. To make this more efficient, we can make use of the Hadoop Java API to convert the XML files to tab-delimited and move the delimited output directly into HDFS in one step.

The remainder of the information in the mzML file is still important to the complete understanding of the outcome of an experiment. A hybrid-load architecture was discussed in previous work by Hillman,[8] where metadata was stored in a normalized schema in a relational database and the scan information in a file system. However, the metadata is not discussed in any further detail in this article.

2D peak picking

In order to pick out the peaks in each scan, a Map task is needed. Here the key/value input is created by a standard file input task with the key being the byte offset into the file and the value being the text from the file up to the next newline character. This is where the tab-delimited file format greatly simplifies processing, as each record in the file up to a carriage return represents a single scan from the mass spectrometer. In this way each scan can be processed completely independently of the others, and therefore the 2D peak-picking process can be made to operate in parallel. As Hadoop is a linearly scalable architecture, it is also a simple matter to time how many scans can be processed in a time frame and calculate how many data nodes will be needed to process a complete experiment in a desired period. The Map task decodes the Base64 binary arrays storing the mass-to-charge and intensity data and loads them into Java arrays in memory. Each peak is detected by using a slope detection algorithm. Overlapping peaks introduce some degree of complexity here. In addition, noise in the signal and the way the instrument measures the peptides mean that the peaks can be shifted slightly; however, it is possible to compensate for this by calculating a theoretical peak by fitting an ideal Gaussian curve to the data (Fig. 3).

"THE REMAINDER OF THE INFORMATION IN THE MZML FILE IS STILL IMPORTANT TO THE COMPLETE UNDERSTANDING OF THE OUTCOME OF AN EXPERIMENT."

De-isotoping 2D peaks

Because of the presence of carbon isotopes, it is necessary to identify peaks within an isotopic envelope that represent the same peptide. This can be done in the same Map task as the 2D peak picking. As the scan data is looped over and the weighted theoretical peaks identified, the mass-to-charge ratio, intensity, and other relevant information is stored in an array. Once a first pass over, the scan is completed, and the weighted peak array can be processed to look for isotopic peaks. This is carried out by joining the array back on itself with an inner and outer loop. As peaks are matched within an isotopic window, the charge can be calculated, thus completing the first step in the parallel-processing workflow.

3D peak picking

The 2D peaks identified so far are indicators of the presence of a peptide. As stated above, the mass spectrometer carries out multiple scans because any one peptide can take several seconds to pass through the machine. Hence, the same peptide appears as multiple 2D peaks in multiple scans. By treating these multiple peaks as a single 3D peak and calculating the volume under that peak, it is possible to calculate

the quantity of the peptide in the sample. Once the 2D peaks have been identified from the individual scans, the dataset is greatly reduced. Where the original scan may have contained 20,000 data points, the total number of 2D peaks could number below 2000. This reduction in data is important, as now that the first Map step is complete, the data will need to be redistributed around the cluster and written out to disk. The redistribution is performed by the shuffle step of the MapReduce process as it directs the Map output key and value pairs to the correct reducer. The 3D peaks are detected by looking across the scans and matching 2D peaks within a mass and a retention time window. These windows will need to overlap to ensure that all peaks are detected, and this is also handled by the output from the 2D peak Map task. A custom partitioner function is required to ensure that the output of the initial 2D peaking process is ordered correctly by the mass of the peak, the scan number, and the retention time and to allow the overlap window. The reason for this is that a similar algorithm that detected the 2D peaks can now be used to detect the 3D ones across the scans. This step is a Reduce step as the 3D peaks will occur within a mass window (chosen to be 7 ppm in this work). In this way, the 2D and 3D peak-picking process fits well into the MapReduce programming framework, and where data needs to be redistributed, the dataset has been greatly reduced by the Map Task.

"IN THIS WAY, THE 2D AND 3D PEAK-PICKING PROCESS FITS WELL INTO THE MAPREDUCE PROGRAMMING FRAMEWORK, AND WHERE DATA NEEDS TO BE REDISTRIBUTED, THE DATASET HAS BEEN GREATLY REDUCED BY THE MAP TASK."

3D isotopic envelopes

In the same way as described above for the 2D peaks, the 3D peaks also require a de-isotoping step. To do this, the 3D peaks can again be stored in an array and the same techniques used to identify peaks that have different masses but, because of the presence of carbon isotopes, are actually the same peptide. At this point we have calculated the mass and intensity of molecules and can either output the results or move on to further processing such as detection of stable isotope labeling by amino acids in cell culture (SILAC) or database search.

Evaluation

As stated above, the 2D and 3D peak-picking process fits very well into the MapReduce framework. Each scan produced by the mass spectrometer can be processed independently by a Map task. This Map task can handle complexity such as decoding the Base64 arrays, removing noise in the signal, and handling overlapping peaks. The 3D peak picking follows a similar pattern to the 2D but in a Reduce task that needs to look across scans in an overlapping mass window. Timings have been calculated on the 2D process for comparison with a desktop PC-based process that uses MaxQuant software to do the processing.

Using a test Hadoop cluster running Hadoop in Virtual machines on a Windows host machine, timings were taken for the 2D peak-picking process. The timings were compared using 2, 3, and 4 data nodes with each node having 2 Map slots. Even on this small scale, the linear scalability of Hadoop can be seen with a reasonably consistent number of scans per second per Map task. The completion times were impressive compared with a conventional PC-based process, which completed the same task in around 22 minutes. Using the metric of 25 scans per slot per second, a Hadoop cluster containing 10 data nodes with 16 Map slots would be able to process 4,000 scans per second.

Discussion/Future Work

There are many areas still to be researched in this process, including the SILAC pair/triplet detection and, importantly, the database search that identifies the peptides by their mass and ties the peptides to a given protein.[9] A complete version of the process coded in the MapReduce framework will allow timings to be taken and compared across platforms and Hadoop configurations. This will also allow direct comparison with the desktop computer processing that is currently carried out. Hadoop open-source software is constantly being updated and added to as the community seeks to expand and improve on its capabilities. Already in the beta stages of release, developments such as YARN, Tez, and Impala are promising improvements in speed, usability, or both. A properly designed and researched process will allow future work to take advantage of technical developments without having to revalidate and redesign the methodology for processing raw mass spectrometer data into actionable information.

Acknowledgment

This work would not have been possible without the support of the Lamond Laboratory at the University of Dundee.

Author Disclosure Statement

No conflicting financial interests exist.

References

1. Palzkill T. Proteomics. New York: Springer, 2002.

2. Dean J, Ghemawat S. MapReduce: simplified data processing on large clusters. Commun ACM 2008; 51:107–113.
3. Surajit C, Umeshwar D, Narasayya V. An overview of business intelligence technology. Commun ACM 2012; 54:88–98.
4. Martin K, Yuriy K, Siegfried B, et al. A cloud framework for high throughput biological data processing. Presented at the International Symposium on Grids and Clouds and the Open Grid Forum, Taipei, Taiwan, 2011.
5. Lewis S, Csordas A, Killcoyne S, et al. Hydra: a scalable proteomic search engine which utilizes the Hadoop distributed computing framework. BMC Bioinformatics 2012; 13:324.
6. Mohammed Y, Mostovenko E, Henneman AA, et al. Cloud parallel processing of tandem mass spectrometry based proteomics data. J Proteome Res 2012; 11:5101–5108.
7. Deutsch E, Martens L. et al., mzML: mass spectrometry markup language. 2009. http://www.psidev.info/index.php?qnode/257
8. Hillman C. Investigation of the extraction, transformation and loading of mass spectrometer RAW Files. MSc thesis. University of Dundee, United Kingdom. 2012.
9. Gevaert K, Impens F, Ghesquière B, et al. Stable isotopic labeling in proteomics. Proteomics 2008; 8:4873–4885.

Address correspondence to:

Chris Hillman
School of Computing
University of Dundee
Nethergate
Dundee DD14HN
United Kingdom

E-mail: c.hillman@dundee.ac.uk